LSI

Rahul Garg(2003CS10183) Varun Gulshan(2003CS10191)

October 29, 2005

1 Overview of Approach

Let D be terms by document matrix. We have to broadly handle two problems namely

1. Synonymy
2. Polysemy

SVD decomposition can discover the latent semantics hidden inside the matrix D.Consider the SVD decomposition of D.

$$D = U\Sigma V^T$$

Consider the k-rank approximation of D i.e

$$\hat{D} = \hat{U}\hat{\Sigma}\hat{V^T}$$

The value of k should be chosen carefully and it should not be less than the no of categories of documents. The main aim of doing a svd is to cluster similar documents. Though the documents are close together in n dimensional space also but reducing them to two dimensions helps in solving the problem of synonymy and polysemy.

1.1 How synonymy is solved

Consider two categories of documents **cs** and **maths**. A term(say t) which belongs the category cs but does not appear in some cs document(lets call it d_{cs}) but by chance in some maths document(lets call it d_{maths}),when compared with d_{cs} in the original dimension is orthogonal to it hence the document d_{cs} does not appear in the results at all,whereas the document d_{maths} turns up.But taking the projection of the query into a reduced space brings the document d_{cs} and the term t closer because they both have components along the singular vector corresponding to cs documents. Also the document d_{maths} is away from t in the reduced space because it has its major component along the singular vector corresponding to maths documents and is almost orthogonal to the query corresponding to the term t.

1.2 How polysemy is solved

Consider similar setup as in previous case,i.e two categories of documents cs and maths. if the term(say t) is a term belonging to both maths and cs documents then in the reduced space a query of only the term t will lie exactly midway between the two axis corresponding to cs and maths terms. But suppose along with it we have another term which belongs only to cs documents,then the query vector will move closer to the cs axis and hence the meaning of the ambiguos term is resolved by the other terms. Though the same argument can apply even when we dont do a rank reduction,the problem in the original dimension is the same as the problem that happens with synonymy,some documents that should show up do not,and some other wrong documents turn out.

1.3 Representation of documents in reduced dimensional space

Documents have now been projected to a new reduced dimensional space and their coordinates in the new space can be obtained using

$$Documents = \hat{U^T} D$$

Conider a query Q. It will be a column vector having 1's corresponding to search terms. We can treat it as a pseudo document. To compare it with other documents, we first project it onto the reduced dimensional space

$$\hat{Q} = \hat{U^T} Q$$

Now the query can simple be compared using dot product and comparing the angles between document vectors.

1.4 Representation of term vectors in reduced dimensional space

Like the document vectors, coordinates of row vectors may be obtained using

$$Terms^T = D\hat{V}$$

2 Implementation and Discussion

Implementation essentially consists of the approach outlined above. But their are a few minor issues that need to be handled like normalizing the Document matrix before taking the SVD decomposition, normalization of Query,etc. Note that if we don't normalize the Document matrix then a long document can bias the SVD projection towards itself. Following matlab function have been implemented

1. plotDocuments
   Shows a 2D plot of document vectors projected onto the 2 dimensional space.

2. plotTerms
   Shows a 2D plot of term vectors projected onto the 2D space

3. convToQuery
   Takes an input string and returns the corresponding query vector.

4. matchDocuments
   Takes query vector and document matrix and dimensions to reduce to and returns a row vector containing the cosines of the angles which the query vector makes with the document vectors in the reduced dimensional space.It also displays the 2D or 3D plots if number of dimensions is 2 or 3.

5. plotDocuments3D Plots the documents in 3D.

6. matchQR It is similar to matchDocuments but it works using QR approach. It also shows the 2D plot if number of dimensions is 2.

In addition to matlab code, following c++ code has also been implemented

1. generate
   read file 'title.txt' and outputs file 'terms.txt' containing the various terms in the titles read

2. matrix
   read files 'title.txt' and 'terms.txt' and dumps the terms list and document matrix.

2.1 Examples

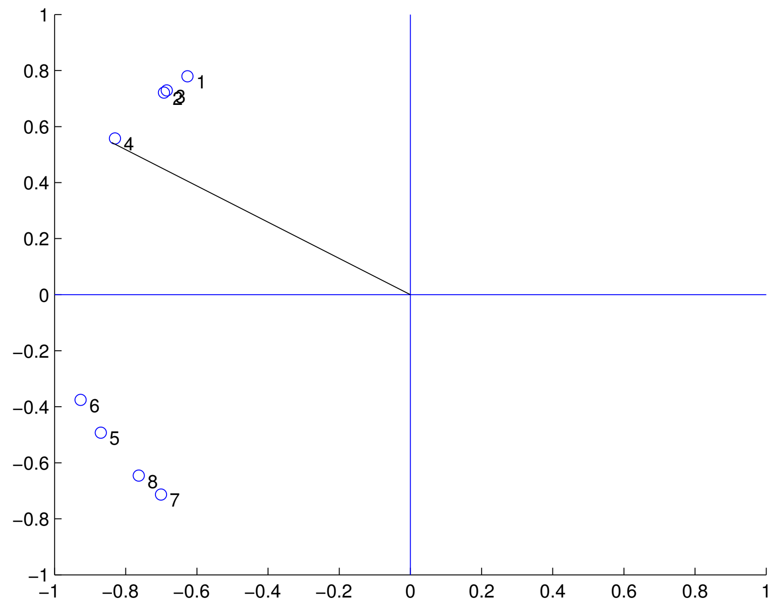
This approach can be analysed by studying the following examples

1. The first example is a manually constructed example. The document term matrix looks like

			cs				ma		
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	
t_1	1	1	1	0	0	0	0	0	**cs**
t_2	1	1	1	1	0	0	0	0	**cs**
t_3	0	1	1	1	0	0	0	0	**cs**
t_4	1	1	0	1	0	1	0	0	**cs**
t_5	0	1	1	0	1	1	0	0	**cs and maths**
t_6	0	0	0	0	1	1	1	0	**maths**
t_7	0	0	0	0	1	1	1	1	**maths**
t_8	0	0	0	1	1	1	0	1	**maths**
t_9	0	0	0	0	0	1	1	1	**maths**

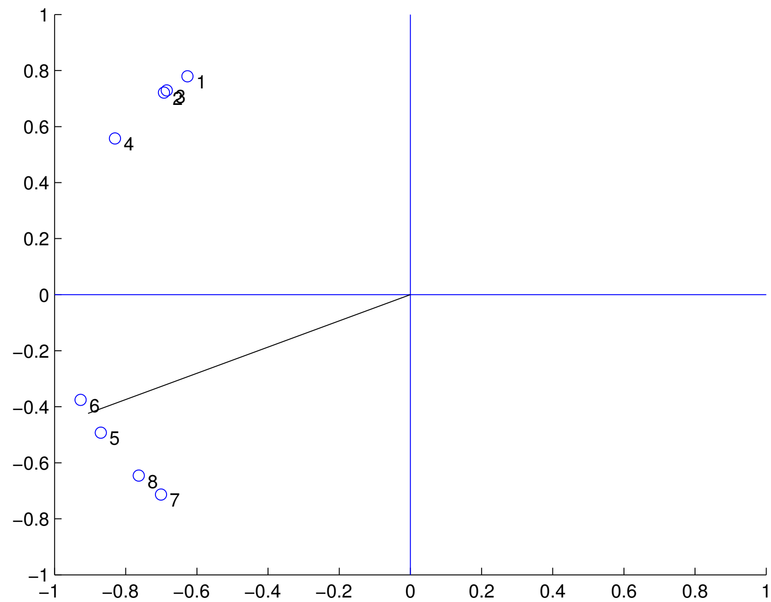
   Here we have two categories of documents , lets call them computers and maths. The first four terms are related to computers and the last four related to maths. The fifth term is a common term appearing in both the documents. This term can give rise to polysemy. If it is qualified with another term which appears in computers terms, then this method throws out the computers documents before the maths documents. We can consider examples where plain text matching would give wrong results

Polysemy example: Consider a search with terms t_4 and t_5. The documents d_6 textually matches the query more than d_4 or d_3 or d_1,but is not correct. Our plot cleary shows that the correct documents d_1 d_2 d_3 d_4 are closer to the query than the other documents.



In this plot we have reduced the documents and the query to two dimensions and normalized them. The dots represent the documents and the query is represented by the tip of the line.So the angle between the query and the document is a measure of the match between them.

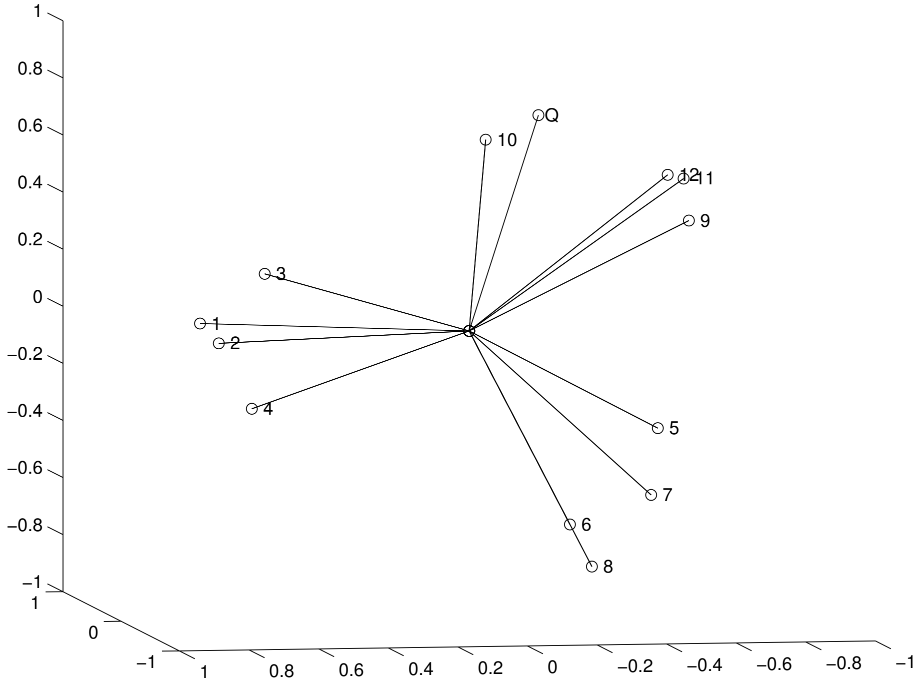
Synonymy: Consider a search with the term t_8. This term does not appear in d_7 but appears in d_4. Simple text matching would report d_7 before d_4 but from the figure clearly d_7 is closer to the query.



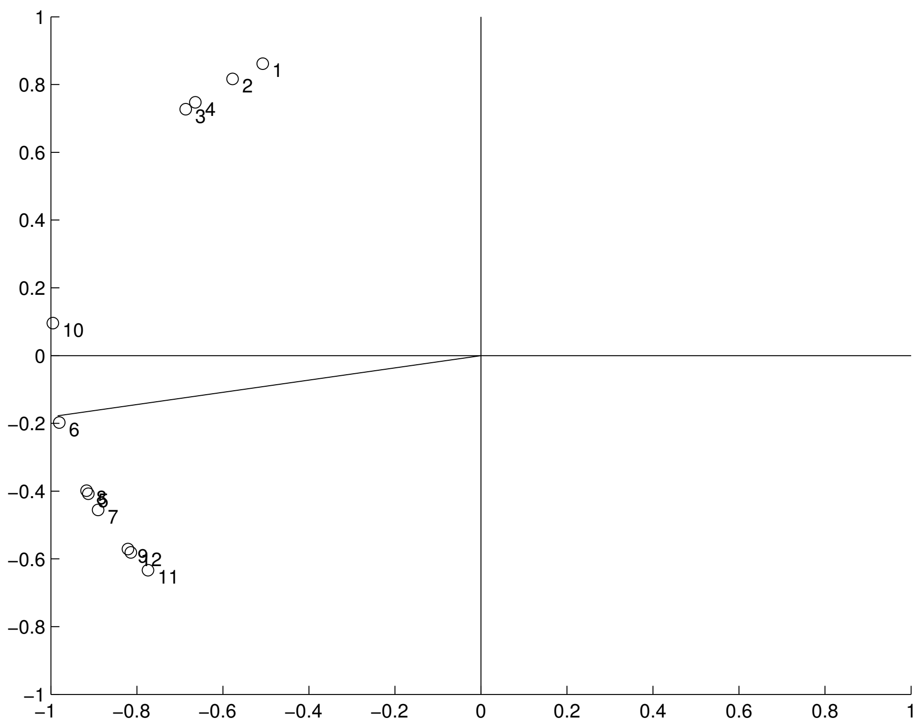
2. The next example is also manually constructed but has three categories lets call them cs, maths and ee. Its document term matrix looks like

			cs				ma				ee			
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8						
t_1	1	1	1	0	0	0	0	0	0	0	0	0	**cs**
t_2	1	1	1	1	0	0	0	0	0	1	0	0	**cs**
t_3	0	1	1	1	0	0	0	0	0	0	0	0	**cs**
t_4	1	1	0	1	0	1	0	0	0	0	0	0	**cs**
t_5	0	1	1	0	1	0	1	0	0	0	0	0	**cs and ma**
t_6	0	0	0	0	1	1	1	0	1	0	1	1	**ma and ee**
t_7	0	0	0	0	1	1	1	1	0	0	0	0	**ma**
t_8	0	0	0	1	1	1	0	1	0	0	0	0	**ma**
t_9	0	0	0	0	0	1	1	1	1	0	0	0	**ma**
t_{10}	0	0	0	0	1	0	0	0	1	1	1	1	**ee**
t_{11}	0	0	1	0	0	0	0	0	1	1	0	1	**ee**
t_{12}	0	0	0	0	0	0	0	0	1	0	1	1	**ee**
t_{13}	0	0	0	0	0	0	0	0	1	1	1	0	**ee**

Synonymy example: Here also consider the term t_{11},which is an ee term but appears in d_3 also , but the plot shows it is closer to the ee documents.

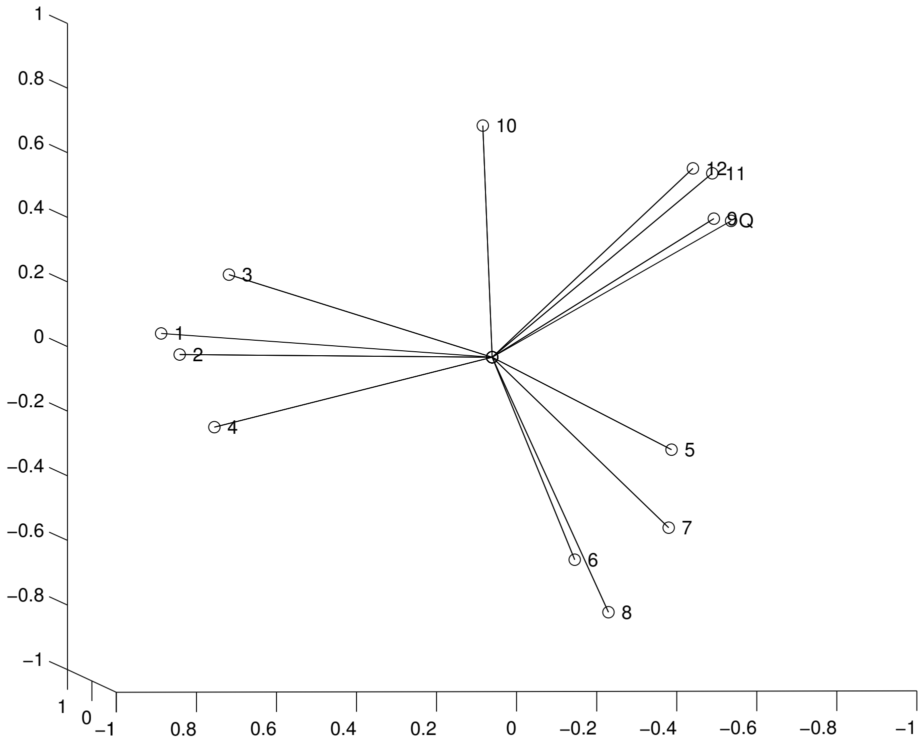


The point Q denotes the query. However query fails if we work with 2 dimensions, can be seen in this plot



In 2 dimensions the query is going to turn out maths documents before.

Polysemy example: The term t_6 is common to both maths and ee, but the query of the terms t_6 and t_{10} gives out the ee documents because of the term t_{10}.



3. The next example considers the following books as its database

    (a) Linear Algebra : Solving System of Equations
    (b) Computing roots of system of equations using algebra
    (c) System of Linear Equations
    (d) Roots of System of Equations
    (e) Algebra : solving system of numerical equations
    (f) Algorithms to find roots of system of linear equations
    (g) Control System and related Algorithms
    (h) Control System and Numerical and Scientific Algorithms
    (i) Numerical and Scientific Computing
    (j) Control Algorithms and Computing
    (k) Control System and Numerical Algorithms

    The underlined word show the terms used for indexing. First 6 documents relate to maths while latter documents relate to CSE. Hence on a 2D plot

we'll like to these 2 categories separate out. Note that *system* is a term which shows polysemy.
Now consider the searches *system equations* and *control system*. We would like the first query to return documents related to math while the second query should return CS documents.Following diagrams show the two queries respectively

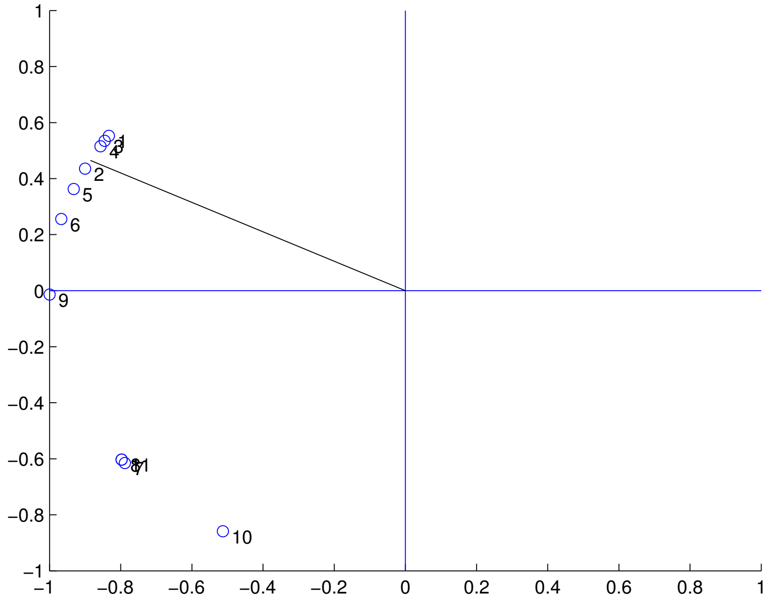


Figure 1: Query: system equations. Clearly the query is near the maths documents which are themselves clustered together

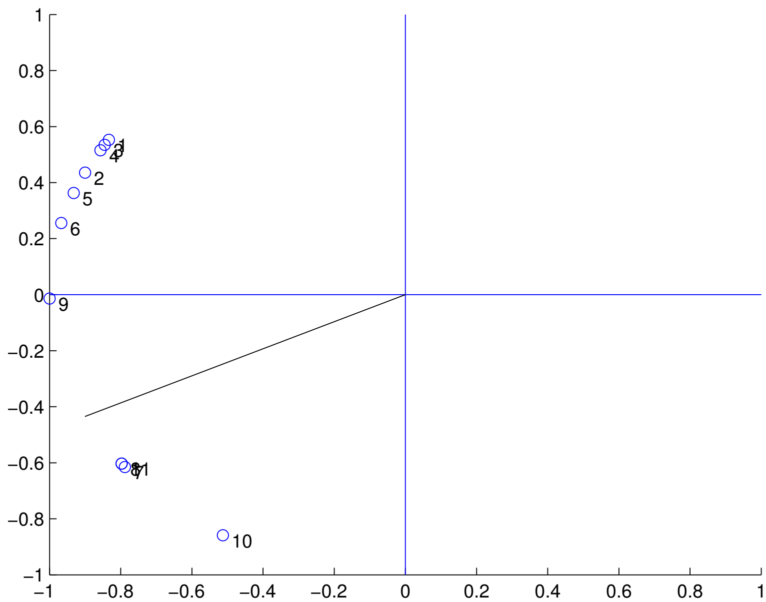


Figure 2: Query: control system. Here the query is near the CS documents

Clearly, we have been able to infer the correct meaning of polysemic term *system* with the help of other search terms.
This example may also be used to demonstrate the fallacies of QR based approach. Intuitively, this approach should not work since we drop q's arbitrarily and we may end up dropping directions of maximum variations. To show this, we again consider the same queries but with QR approach.Following plots are obtained

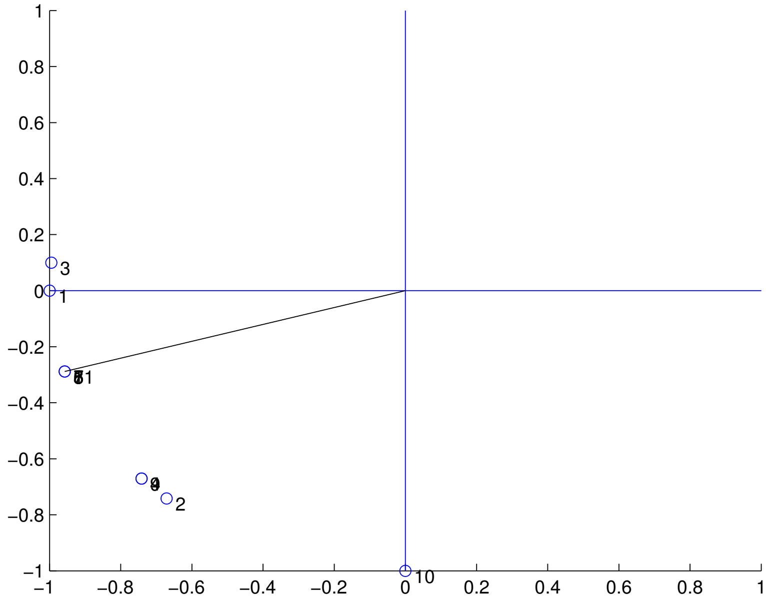


Figure 3: Query: system equations. Clearly the query is nowhere near the math documents. In fact the math documents ar themselves scattered

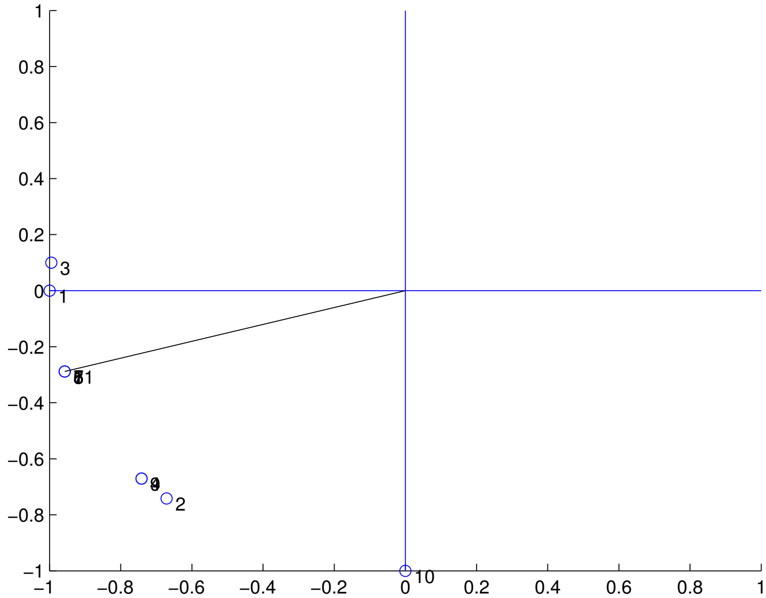


Figure 4: Query: control system. Note that this query is no different from the previous query. Hence the system is incapable of distinguishing between such queries

We can improve the QR by considering those Q's such that there is minimum loss of variation. We can remove the Q's whose corr row in the matrix R is of the least norm. QR does not work because we are not changing the basis to represent the documents in a sensible way. We should change the basis such that the first k coordinates(k is the rank approximation) represent the documents as closely as possible, but QR does not to do that. Also QR factorization depends on the order in which we place the documents in the matrix, hence it may give good vectors some time and bad vectors some times.